Application of Adaptive Probing for Fault Diagnosis in Computer Networks [1]

Maitreya Natu
Dept. of Computer and Information Sciences
University of Delaware, Newark, DE, USA, 19716
Email: natu@cis.udel.edu

Adarshpal S. Sethi
Dept. of Computer and Information Sciences
University of Delaware, Newark, DE, USA, 19716
Email: sethi@cis.udel.edu

***Abstract*— This dissertation presents an adaptive probing based tool for fault diagnosis in computer networks by addressing the problems of probe station selection and probe selection. We first present algorithms to place probe stations to monitor the network in the presence of various failures in the network. We then present algorithms for probe selection in an adaptive manner to perform fault diagnosis. We present algorithms considering both deterministic as well as non-deterministic environments. We present evaluation of the proposed algorithms through comprehensive simulation studies. The dissertation is available at http://www.cis.udel.edu/∼natu/papers/dissertation.pdf.**

I. Introduction

Fault localization is the process of diagnosing the cause of failure from the observed failure indications. With the increasing size and complexity of computer networks and with the increasing demands for quality of service guarantees in network services, fault localization has become an important network management task. Existing fault localization techniques are inadequate to address various challenges introduced by modern communication systems. A fault diagnosis solution for modern communication systems should have the following properties:

- **Ability to perform reasoning under uncertainty about the underlying dependencies:** In many scenarios, for instance in case of dynamic routing, complete and accurate information about a network's topology, routes, participating nodes, etc., is unknown.
- **Diagnosis of scenarios with multiple failures:** With the increasing size and complexity of networks, and with the development of large multi-component enterprise systems, the likelihood of multiple failures in the system increases.
- **Diagnosis with low management traffic overhead:** The management traffic should be minimized so that performance of active network applications is not adversely affected because of network management overhead.
- **Small deployment cost:** It should be possible to deploy diagnosis tools over a network with small instrumentation overhead.
- **Quick localization:** To minimize the damage caused by a failure, it is important to localize the fault quickly, and take immediate corrective actions.
- **High accuracy:** It is important to perform accurate fault diagnosis and generate fewer false positives.

This dissertation [2] presents techniques that allow fault localization to be performed in an automated manner using adaptive probing. Probing-based tools send probes in the network to analyze the health of network components, where probes are test transactions designed such that success or failure of a probe depends on the success or failure of the network components visited by the probe. Example of probes are pings, trace-routes, HTTP requests, etc. In the past, preplanned probing has been used which involves designing a preplanned set of probes that is capable of diagnosing all possible failure scenarios of interest and sending this set of probes periodically in the network. Adaptive probing on the other hand, proposes to adapt the probe set to the observed network conditions by sending more probes in the suspected areas and less probes in the healthy areas of the network. Adaptive probing involves lower management traffic and provides faster and more accurate fault localization.

With adaptive probing, we attempt to meet the above stated requirements of a fault diagnosis tool. We provide adaptive probing solutions assuming the availability of deterministic and non-deterministic dependency information. We attempt to find multiple failures; however, we assume a limit on the maximum number of failures that can be diagnosed. Adaptive probing attempts to minimize the overhead of probe traffic by analyzing the network state from previous probe results, and identifying new probes to send that can provide maximum information. Probing-based fault diagnosis solutions involve low instrumentation overhead requiring only the instrumentation of probe stations. Adaptive probing being incremental and adaptive in nature is computationally less complex than the preplanned approach of probe selection. Also, we show through simulation results that adaptive probing provides a high detection ratio and low false positive ratio. The proposed techniques can be applied in a variety of domains such as monitoring of telephone networks, performance monitoring in e-Commerce systems, diagnosis of link and node failures, etc.

[1] Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

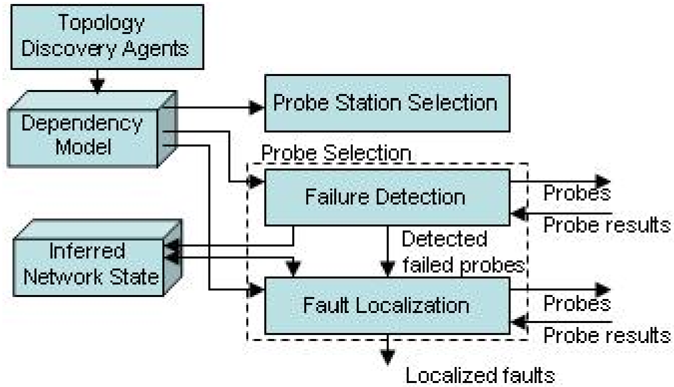

Fig. 1. System architecture for fault diagnosis using adaptive probing.

In the algorithms proposed in this dissertation we perform localization of node failures in a network.

II. Adaptive probing system architecture

Figure 1 presents the components of the proposed system architecture for an adaptive probing tool for fault diagnosis. Below we describe these components:

- *Probe Station Selection:* Probe stations are the nodes responsible for sending and analyzing probes. Probe Station Selection component finds suitable locations in the network where probe stations should be deployed such that the probes could be sent to monitor the entire network for faults even in the presence of multiple failures. The number of probe stations should be small to reduce the deployment overhead.
- *Probe Selection:* Probe Selection component selects suitable probes to be sent over the network. Using adaptive probing, we divide the probing task into two sub-tasks which we call *Failure Detection* and *Fault Localization*.
  - The *Failure Detection* module periodically sends a small number of probes such that the failure of any managed component can be detected. These probes are few in number, and are designed to only detect the presence of a failure in the network. They might be unable to exactly localize a failure.
  - Once a failure is detected in a network, the *Fault Localization* component analyzes probe results, and selects additional probes that provide maximum information about the suspected area of the network. This process of probe analysis and selection is performed to localize the exact cause of failure.
- *Topology Discovery Agents:* The Probe Station Selection and Probe Selection modules use the dependencies between probes and nodes on the probe path. This dependency information is obtained through *Topology Discovery Agents*, and is stored in a dependency model. The collection of dependency information could be done in a variety of ways. For instance, the dependency between the end-to-end network probe path and the network nodes can be obtained using traceroutes or the topology tables of the underlying network layer.
- *Dependency Model:* This model stores the relationship between probes and nodes. It stores the information about the nodes that are visited by a probe, so that the health of nodes can be inferred from the success or failure of the probes. The dependency model could be deterministic when complete and accurate information is available about the probe paths. For situations in which complete and accurate information about the probe paths is unavailable, a probabilistic model is used where the dependency between probe paths and nodes is represented using the probability of causal implication of a node failure on a probe failure. The causal probabilities can be computed in a variety of ways. For instance, in a scenario of multi-path routing, the probabilities could be based on the policy that the routers or load balancers use to select the next hop. In this dissertation, we consider scenarios of deterministic as well as probabilistic dependency models.
- *Inferred Network State:* The Failure Detection and Fault Localization components of the architecture store intermediate diagnosis results in the Inferred Network State. These results are refined by the Fault Localization component by sending and analyzing more probes.

III. Problems addressed in this dissertation

An important problem that needs to be addressed while developing adaptive probing based solutions is the selection of network nodes on which *probe stations* should be placed. Various design issues need to be considered while developing probe station selection algorithms, such as the presence of multiple failures, availability of the dependency information, and nature of failures. Probe station selection algorithms proposed in the past suffer from two main limitations:

- Most existing work proposes placement of probe stations to monitor all components of interest to obtain various performance metrics and fails to consider the failure of network components, making the solutions vulnerable to such failures. Most existing work also fails to consider probe station failures.
- Most existing probe station selection algorithms assume the availability of a deterministic and complete dependency model, which make them impractical in various real-life scenarios.

Another problem that we address in this dissertation is the selection of *probes*. Past work on probe selection has certain limitations:

- Most probing-related research in the past is based on preplanned probing. Adaptive probing is more suitable because of its lower management traffic, and faster and more accurate localization.
- Probe selection algorithms presented in the past assume a deterministic environment with complete and accurate information about the network. Analysis of success and failure of probes to infer health of network components becomes difficult in a non-deterministic environment.

IV. Contributions of this dissertation

This dissertation makes the following specific contributions to the field of failure detection and fault localization:

- Proposes algorithms for placing probe stations in a network to monitor node health and diagnose node failures. The algorithms address the limitations of existing probe station placement algorithms by considering the possibility of node failures as well as probe station failures in the network. We present two algorithms: Algorithm SNR and Algorithm SNR-PSF making different assumptions about node failures and probe station failures. This work has been published in [7], and has been submitted to [8].
- Proposes a heuristic-based approach in Algorithm GFD for probe selection for failure detection, and show through experimental evaluation that the proposed algorithm provides better results than algorithms proposed in the past. This work has been published in [3].
- Proposes novel algorithms, Algorithm GFL using MinSearch, Algorithm GFL using MaxSearch, and Algorithm BSFL, to perform fault localization by sending probes in an adaptive manner for a deterministic environment. This work has been published in [4] and has been submitted to [8].
- Proposes novel adaptive probing solutions for probe station selection and probe selection for a non-deterministic environment using a probabilistic dependency model. Existing probing research assumes a deterministic environment and fails to address the real-life uncertainties existing in the system that appear during actual system deployment. We address these uncertainties and propose the algorithms for probe station selection and probe selection in the presence of incomplete and inaccurate information about the underlying environment. We present Algorithm PSNR for probe station selection, Algorithm PPFL for preplanned probe set selection, and Algorithms PGFD and PGFL for adaptive probe selection in a non-deterministic environment. We further demonstrate the effectiveness and efficiency of adaptive-probing-based tools over preplanned-probing-based tools through experimental evaluation. This work has been published in [5], [9] and has been submitted to [6].

A copy of the dissertation is available at [2].

V. Probe station selection

We propose a heuristic-based approach that incrementally selects nodes which provide suitable locations to instantiate probe stations. The proposed algorithms involve fewer computations than the combinatorial approach. These algorithms attempt to find the minimal probe station set but are not guaranteed to do so. We show through simulation results that these algorithms give results that are close to optimal. We first assume a deterministic environment representing the dependencies between the probes and the nodes using a deterministic dependency model. We later relax the assumptions on the available dependency information and consider a non-deterministic environment.

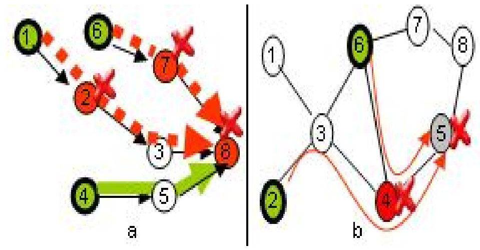

Fig. 2. (a.) 3 independent paths to node 8 from probe station nodes 1, 6, and 4, to detect failure of node 8 in a scenario of failure of 3 nodes (nodes 2, 7, and 8); (b.) A scenario of failure of nodes 4 and 5, where inappropriate probe station placement (at nodes 2 and 6) makes node 5 a shadow node;

A. Design rationale

It can be proved that with the presence of at most k failures in the network, a set of probe stations can localize any k non-probe-station node failures in the network if and only if there exist k independent probe paths to each non-probe-station node.

Consider the example shown in Figure 2a. This figure shows how the availability of three independent paths to node 8 from probe station nodes 1, 4, and 6, helps to diagnose the failure of node 8 in a scenario of three node failures (nodes 2, 7, and 8). Failure of nodes 2 and 7 prevent probe stations 1 and 6 respectively from diagnosing the health of node 8. However, with the assumption of diagnosing at most three faults, and the availability of three independent paths, there is one probe path to node 8 (path from 4 to 8) with no intermediate failed nodes. Thus probe station node 4 can detect the failure of node 8.

An incorrect probe station placement can make some nodes unreachable in certain failure scenarios and thus make these nodes *shadow nodes*. Figure 2b shows how an incorrect probe station placement at nodes 2 and 6 can lead to inadequate diagnostic power to detect failure of nodes 4 and 5. This figure shows paths from nodes 2 and 6 to reach node 5. It can be seen that the failure of node 4 makes node 5 unreachable from both the probe stations, making node 5 a shadow node.

The probe station selection problem can then be defined as

> *Select the smallest set of nodes as probe stations such that every node that is not a probe station has k independent paths from the probe stations.*

In the dissertation, we prove that the Minimum Probe Station Selection problem is NP-Complete by reducing the Minimum Set Cover problem to the Minimum Probe Station Selection problem. We then present following heuristic-based algorithms for probe station selection:

1) Algorithm SNR: Probe station selection algorithm considering node failures and assuming no probe station failures.
2) Algorithm SNR-PSF: Probe station selection algorithm considering both node failures and probe station failures.
3) Algorithm PSNR: Probabilistic probe station selection algorithm considering node failures and assuming no probe station failures.

B. Probe station selection in a non-deterministic environment

While selecting probe stations in a non-deterministic environment, because of the involved probabilities, the independence of the two paths p and q cannot be declared with absolute certainty and hence needs to be represented with a certain belief value. This value represents the confidence in our belief that the two paths are independent. The belief value can be computed as follows:

$$B(I_{(p,q)}) = 1 - P(\bigcup_{n \in Nodes(p) \cap Nodes(q)} R_n(p,q)) \quad (1)$$

where the term $R_n(p,q)$ represents the event that probes p and q both pass through node n, and $P(R_n(p,q)) = P(p,n) * P(q,n)$. The higher the value of $B(I_{(p,q)})$, the stronger is the belief that probes p and q are independent.

For a candidate probe station c, the probability that c provides a path to a shadow node s that is independent of the paths to s from already selected probe stations in S can be obtained by computing

$$B(I_{(Path(S,n),Path(c,n))}) = 1 - P(\bigcup Q_m(Path(S,n), Path(c,n))) \quad (2)$$

where $m \in Nodes(Path(S,n)) \cap Nodes(Path(c,n))$. The term $Q_m(Path(S,n), Path(c,n))$ represents the event that $Path(S,n)$ and $Path(c,n)$ both pass through the node m and $P(Q_m(Path(S,n), Path(c,n))) = P(Path(S,n), m) * P(Path(c,n), m)$.

If this value is greater than a threshold, then the path is considered to be independent. A candidate node that provides maximum number of independent paths is selected as the next probe station. Once a probe station c is selected, it is added to the set S and the probability $P(Path(S,n), m)$ for each node n to which node c provides an independent path and for each node m used by these paths, is updated as follows:

$$P(Path(S \cup c, n), m) = P(\bigcup (Q_m(Path(S,n), Path(c,n)))) \quad (3)$$

VI. Probe Selection

We define the problems of probe selection for *failure detection* and *fault localization* and prove them to be NP-Complete. We then present heuristic-based algorithms to select probes to perform *failure detection* and adaptive probe selection algorithms to perform *fault localization*. We first assume a deterministic environment representing the dependencies between the probes and the nodes using a deterministic dependency model. We later relax the assumptions on the available dependency information and consider a non-deterministic environment. We also present a preplanned probing algorithm where a set of probes is selected to localize all possible faults in the network and is sent periodically over the network. Preplanned probing involves a high computational complexity and we show through simulation results that preplanned probing is less accurate and requires much larger number of probes as compared to adaptive probing.

We present the following algorithms:

1) Algorithm GFD: Algorithm for probe selection for failure detection. We compare this algorithm with the Additive algorithm presented by Brodie et. al. [1] through experimental evaluation.
2) Algorithm GFL: Algorithm for adaptive-probing-based probe selection for fault localization using two approaches: Min Search and Max Search.
3) Algorithm BSFL: Algorithm for adaptive-probing-based probe selection for fault localization, where probes are selected to test failed probe paths in a binary search fashion.
4) Algorithm PPFL: Probabilistic preplanned-probing-based probe selection algorithm.
5) Algorithm PGFD: Probabilistic probe selection algorithm for failure detection.
6) Algorithm PGFL: Probabilistic adaptive-probing-based probe selection algorithm for fault localization using two approaches: Min Search and Max Search.

A. Failure detection

Probes for failure detection should be selected such that, in the presence of a fault in the network, some of the selected probes should fail, causing the detection of the failure by the network manager. As the probes for failure detection are sent at periodic intervals, they should be optimized to prevent overwhelming the network resources and affecting the performance of other applications using the network. The probe set selection problem for failure detection can then be defined as:

> *Given a set of probes and given the dependency information between the probes and the network nodes, select the smallest set of probes such that every node in the network is covered by some probe.*

In the dissertation, we first prove that the probe set selection for failure detection is NP-Complete by reducing Minimum Set Cover problem to the Minimum Failure-Detection-Probe-Set Selection problem. We then present an approximation algorithm for the selection of such a probe set.

We present a greedy approximation algorithm (Algorithm GFD) that explores the information contained in the dependencies between probes and network components. The algorithm selects the network element n which is probed by the least number of probes, using the dependency information between probes and probed elements. Out of all the probes probing element n, the algorithm selects the probe which goes through a maximum number of nodes that are not yet probed. Probe selection is done till all nodes are covered by the selected probes.

1) Failure detection in a non-deterministic environment: The probe selection criteria for failure detection is based on identifying the nodes where the uncertainty in selection is minimum, and then applying the Greedy approach of selecting a probe that gives maximum coverage of nodes among all the probes that pass through this node. With a probabilistic dependency model, we identify the node with minimum probe selection uncertainty by computing the entropy of the probabilities by which the node is probed by probes. The entropy for a node n is computed as follows:

$$H(n) = \sum_{\forall p|(p \in AvailableProbes)\&(P(p|n)>0)} -P(p|n)log(P(p|n)) \quad (4)$$

where $P(p|n)$ represents the probability that probe p passes through node n. The node with minimum entropy is chosen as the next node to cover.

Once a node n is selected from the set of nodes N, of all the probes that probe this node, the probe that gives maximum coverage is selected. In a non-deterministic environment, two factors decide the probe selection for failure detection: (1) probability gain obtained in covering node n, and (2) probability gain obtained in covering other nodes. For each node m, we maintain a value $Coverage(m)$ to represent the probability that the node m has been covered by the probes selected so far. We select a probe that maximizes the metric to evaluate the improvement that can be obtained in the probability of covering node n and the probability of covering other nodes by selecting a certain probe p. We represent this metric as follows:

$$\begin{aligned} Gain(p) &= (P(p|n) - P(p|n) * Coverage(n)) \\ &+ \sum_{m \in N-\{n\}} (P(p|m) - P(p|m) * Coverage(m)) \end{aligned} \quad (5)$$

B. *Fault localization*

We present algorithms for probe selection for fault localization. We first describe the preplanned approach for probe selection for fault localization and show that the problem of selecting such a probe set is NP-Complete by reducing the Minimum Test Collection problem to the Minimum Fault-Localization-Probe-Set Selection problem. We then present adaptive probing algorithms for fault localization based on different heuristics.

We present a greedy algorithm (Algorithm GFL) to select probes for fault localization. The algorithm maintains sets of failed, passed, and suspected nodes. The set of suspected nodes contains the nodes whose health needs to be determined. This suspected node set is initialized to all nodes that are present on the failed probe paths. The success and failure of the probes sent affect the sets of failed, passed, and suspected nodes. The nodes lying on the paths of successful probes are added to the set of passed nodes and removed from the set of suspected nodes. A node n is declared as failed and added to the failed nodes set when a failed probe goes through a set of nodes such that all nodes other than node n on that path have already been found to have good health. In other words, no other node on that path is present in the suspected node set. In each iteration, the algorithm builds a probe set to be sent over the network to determine the health of the remaining suspected nodes.

We present two approaches to select the probes for probing the nodes in the suspected node set. One approach is to iteratively select a probe that covers maximum number of suspected nodes. The success of such a probe gives a large amount of information by removing all the nodes on that probe path from the suspected node set. However if the probe fails then the probe does not give much information to significantly narrow down the search space of a failed node. Hence another approach could be to select a probe for each suspected node such that it goes through the least number of other suspected nodes. The success of such a probe gives the information about good health of a small number of nodes, reducing the suspected node set only by a small amount. However, the failure of such a probe narrows down the search space significantly.

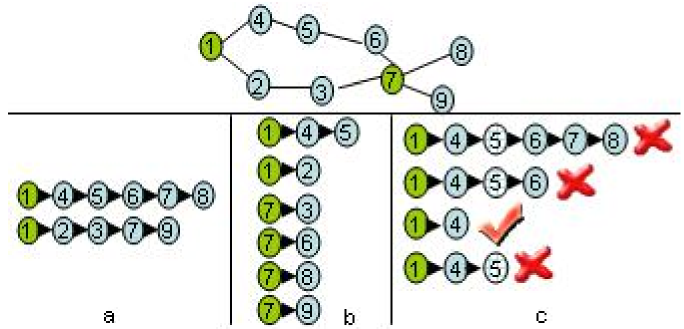

Fig. 3. Network with nodes 1 and 7 as probe stations. (a., b.) Nodes probed by a small set of long probes and larger set of short probes respectively; (c.) Probes sent in a binary search fashion on a failed probe path.

Figures 3a and 3b show an example of how a network can be probed by a set of long and short probes respectively. Success of probe 1→8 gives information about good health of nodes 4, 5, 6, 7, and 8, while its failure narrows down the failure to a set of 5 nodes. This set would need more probes for further diagnosis. On the other hand, success of a smaller probe 1→2 gives little information indicating good health of the single node 2, but failure of this probe narrows the fault localization to a single node, node 2, requiring no more probes for further localization.

The greedy approach does not diagnose each probe path independently. Instead, it builds a single set of suspected nodes consisting of nodes on all failed probe paths. We present a binary search approach, where we propose to diagnose each failed probe path independently. On each failed probe path, additional probes are sent in a binary search fashion till one failure on that path is diagnosed. On a failed probe path, a probe is first sent from the probe station half way on the probe path. If this probe fails, further diagnosis is done on the first half of the probe path. On the other hand, if this probe succeeds, then the later half of the probe path is diagnosed in similar fashion. Figure 3c shows an example of how probes are sent in a binary search fashion to identify a failed node on the probe path. Consider that node 5 has failed and that the failure was detected on observing a failure of probe 1→8. Binary search probe selection then sends probe 1→6. On observing a failure on this path, the first half of the probe path is analyzed focusing on nodes 4, 5, and 6. Continuing probe selection in the binary search fashion, probe 1→4 is sent. Success of this probe indicates good health of node 4, leaving nodes 5 and 6 as the suspected nodes. Next a probe is sent from node 1 to node 5. Failure of this probe together with information about good health of node 4 indicates a failure of node 5.

1) Fault localization in a non-deterministic environment: For a non-deterministic environment, we use a belief metric to express the confidence associated with a given node failure relative to the failure of other nodes. The belief value is initialized to the probability of independent failure of the node, which we represent by $P(n)$. We update this belief value on observing probe successes and failures. The belief computations can be done in an incremental manner. On observing failure of a new probe p, a new belief value can be computed from the previous belief as follows:

$$b_{new}(n) = \beta b_{old}(n) * P(p|n) \tag{6}$$

where $b_{new}(n)$ and $b_{old}(n)$ represent the new and old belief values for failure of node n respectively and β is the normalization constant.

We also take advantage of the fact that some probe failures have not been observed. The fact that many possible probe failures that should occur on a node failure have not occurred should decrease our confidence in the node's failure. On observing a successful probe p, we incorporate the probe success information in the belief computation as follows:

$$b_{new}(n) = \beta b_{old}(n) * (1 - P(p|n)) \tag{7}$$

where β is the normalization constant.

After the analysis of the passed and failed probes and their effect on the belief values of node failures, appropriate probes need to be selected that can give best information for further localization of faults. We propose two approaches for probe selection:

Min Search: For each suspected node s, a probe is chosen that is (1) most likely to pass through node s, and (2) least likely to pass through other suspected nodes. For each probe p under consideration, we compute a metric considering these two factors.

$$probeWorth(p) = P(p|s) + (1 - P(\bigcup_{n \in \{ShadowNodes - s\}} p|n)) \tag{8}$$

where $P(p|s)$ represents the probability that probe p passes through node s, and the term $(1 - P(\bigcup_{n \in \{ShadowNodes-s\}} p|n))$ represents the probability that the probe p does not pass through other suspected nodes. The probe p with maximum value for $probeWorth(p)$ is selected to probe suspected node s.

Max Search: Max search maintains a node coverage probability for each suspected node to indicate the probability that the selected probes pass through the suspected node. The node coverage probability, $nodeCoverage_s$ for each suspected node s is initialized to 0 and is updated with the selection of each probe p as follows:

$$P(p|s) + nodeCoverage_s - P(p|s) * nodeCoverage_s \tag{9}$$

Max search selects the probes that maximize the node coverage probabilities of the suspected nodes. We represent the node coverage obtained by a probe p by the metric $probeGain(p)$ as follows:

$$\sum_{s \in UncoveredNodes} (P(p|s) - P(p|s) * nodeCoverage_s) \tag{10}$$

VII. Conclusion

In this dissertation, we developed tools for fault diagnosis in computer networks using adaptive probing and presented various algorithms for probe station selection and probe selection. Comprehensive simulation studies were done on randomly generated network topologies with sizes up to 500 nodes, and the different algorithms were compared with respect to a variety of metrics. Simulations results were not included for the lack of space. Detailed results are available in the dissertation [2]. With increasing size and complexity of modern communication systems and increasing interest in providing high quality of service, deployment of automated fault localization tools has become critical and is likely to remain an important research area. New fault localization techniques are needed to perform reasoning under uncertainty about underlying system dependencies, deal with multiple failures, create low management traffic overhead, and provide faster diagnosis. The techniques presented in this dissertation advance the field of fault localization by addressing these demands and showing that adaptive probing provides a promising technique for developing fault localization tools for modern communication systems.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.

References

[1] M. Brodie, I. Rish, and S. Ma. Optimizing probe selection for fault localization. In *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Nancy, France*, pages 1147–1157, Oct. 2001.

[2] M. Natu. *Application of Adaptive Probing for Fault Diagnosis in Computer Networks*. PhD dissertation, University of Delaware, Newark, DE, USA, Dec. 2007. http://www.cis.udel.edu/∼natu/papers/dissertation.pdf.

[3] M. Natu and A. S. Sethi. Active probing approach for fault localization in computer networks. In *End-to-End Monitoring Workshop, E2EMON 2006, Vancouver, Canada*, pages 25–33, Apr. 2006.

[4] M. Natu and A. S. Sethi. Efficient probing techniques for fault diagnosis. In *IEEE International Conference on Internet Monitoring and Protection (ICIMP'07), Silicon Valley, CA*, pages 20–25, Jul. 2007.

[5] M. Natu and A. S. Sethi. Probabilistic fault diagnosis using adaptive probing. In *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, San Jose, CA*, pages 38–49, Oct. 2007.

[6] M. Natu and A. S. Sethi. Probabilistic fault diagnosis using adaptive probing. *IEEE Transactions on Network and Service Management, TNSM*, 2007. Submitted.

[7] M. Natu and A. S. Sethi. Probe station placement for robust monitoring of networks. *Journal of Network and Systems Management*, 2007. To appear.

[8] M. Natu and A. S. Sethi. Using adaptive probing for fault diagnosis. *Computer Networks*, 2007. Submitted.

[9] M. Natu and A. S. Sethi. Using adaptive probing for fault diagnosis. In *IEEE GLOBECOM 2007, Washington, D.C.*, Nov. 2007.